Mudd Adventure

Group members: Mingrui Xu, Wei Cao, Shijie Hu, Bowen Dang
Contents

• 1. Overview

• 2. The architectural and timing design

• 3. Difficult parts

• 4. Performance
1.1 Overview

**3D maze Adventure:** The goal of this project is to develop a 3D video game with turn-based 2D fighting system. The player explores the 3D map and the mission of the game is to destroy all the enemies in the map and finally arrive the destination. A gamepad will be the game controller as an input. Also the sound effects will be sent from the speaker during the game.
1.2 Overview

Functionality Graph

System Overview

Software

Hardware

3D display

Mudd Adventure

FSM Game System

Game Pad input

Sound Effect
2.1 The architectural design

Hardware

- Game Module (FSM)
  - Ray Casting Translate
  - SRAM Buffer
  - VGA Raster
  - LED screen

- Texture generator
  - Address Decoder
  - Memory Graphic, Sin cos
  - Texture resample calculation

- USB
- ARM Processor
- Register
- MUX
- Memory (Audio)
- Audio Speaker
2.2 Software
2.3 Timing design

1. FPGA parts works under 50MHz clock. The VGA pixel scanning frequency is 25MHz, and frame frequency should be no less than 25Hz.
2. One Frame time/# of pixels = \(\frac{1}{25/(480 \times 640)} \approx 1.3 \times 10^{-7} \text{ s} \Rightarrow 7.69\text{MHz}\)

3D ray casting block works under 50MHz clock, so every pixel has about 6 clock periods to do transform.
3 Difficult parts

• 1. 64 bit data transformation

• 2. Texture resample calculation

• 3. Timing analysis

• 4. Gameplay based on FSM
3.1 Difficult parts

- 1.64 bit data transformation

  Control signals (gamepad, FSM, random number) (8 bits) + Raycasting results (20 bits) + Texture calculation results (22 bits) = 50 bits!

  Software kernel: 32 bits

  Use twice of order “iowrite32” and two RAMs in the hardware to store them and combine in a 64-bit variable
3.2 Difficult parts

- 2. Texture resample calculation

Graph .mif file: 128*128*24 bits

Two factors to locate the appearance of the graph and scale it with the resample coefficient.

Method: Scan the screen from the bottom

Division operation in software (Rescale the coff)
3.3 Difficult parts

- **3. Timing Analysis**

  Slowest clock frequency: 7.54 MHz

  Noise points all over the screen

  Method: Lower the VGA resolution from 1280 to 640

  Do division calculation in the software

  Use RAMs to store written data
3.4 Difficult parts

- 4. Gameplay based on FSM
In summary, after solving these problems, our game can perform quite well. But there is still some problems with voice output remaining to be solved.

Now let’s see it!